December 2015

Fall Detection by Using Video

Robert J. Gripentog
University of Nevada, Las Vegas

FALL DETECTION BY USING VIDEO

By

Robert John Gripentog

Bachelor of Science – Economics
University of Nevada, Las Vegas
2006

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science in Computer Science

Howard R. Hughes College of Engineering
Department of Computer Science
The Graduate College

University of Nevada, Las Vegas
December 2015

UNLV | GRADUATE COLLEGE

Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

November 24, 2015

This thesis prepared by

Robert Gripentog

entitled

Fall Detection by Using Video

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
Department of Computer Science

Evangelos Yfantis, Ph.D.
Examination Committee Chair

Harlod Bergehel, Ph.D.
Examination Committee Member

Andreas Stefik, Ph.D.
Examination Committee Member

Ying Tian, Ph.D.
Graduate College Faculty Representative

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Abstract

Cameras have become common in our society and as a result there is more video available today than ever before. While the video can be used for entertainment or possibly as storage it can also be used as a sensor capturing crucial information, The information captured can be put to all types of uses, but one particular use is to identify a fall. The importance of identifying a fall can be seen especially in the older population that is affected by falls every year. The falls experienced by the elderly are devastating as they can cause apprehension to normal life activities and in some cases premature death. Another fall related issue is the intentional deception in a business with intent of insurance fraud. Classification algorithms based on video can be constructed to detect falls and separate them as either accidental or intentional. This research paper proposes an algorithm based on frame segmentation, and speed components in the x, y, z directions over time t. The speed components are estimated from the video of orthogonally positioned cameras. The algorithm can discern between fall activities and others like sitting on the floor, lying on the floor, or exercising.

Acknowledgments

I would like to thank Dr. Yfantis for the opportunity to grow as a scientist. Dr. Yfantis' in depth knowledge and grasp of many subjects fit perfectly for my endless questions and inquiries. Love for learning is an inspiration to be around and an aspiration to have. I would also like to thank Dr. Berghel, Dr. Stefik, and Dr. Tian for partaking in my committee. In a life of academia requiring multiple time consuming commitments Dr. Berghel, Dr. Stefik, and Dr. Tian supporting my endeavors makes me truly grateful. I would also like to thank the professors of the computer science department for their assistance and encouragement.

Table of Contents

List of Figures

List of Equations

Chapter 1: Introduction

The elderly population in the United States is growing larger all the time. This growth can be expected to increase as modern medicine and technology continue to extend lifespans. Consequently, age related falls is also increasing and there is a need for assistance to our elderly. Fall detection technology with integrated alert response systems can assist medical workers, assisted living homes, and keep senior citizens independent. Some of the alert response systems require the elder to interact with a device, like pushing a button, to have an emergency responder. Systems that require physical interaction are very accurate but they will also fail if the senior cannot call for assistance, for example if they are unconscious. Thus a system that a senior can truly count on must be able to make an automated decision regarding fall recognition. In general there are two types of devices used to collect data when attempting to automatically detect a fall, those that are worn and those that are not. The problem with devices that are worn like accelerometers is mostly due to comfort, neglect to keep the device charged or on their person. To combat such issues some researchers have suggested to use microphones, floor sensors, or video surveillance. This paper will focus on video surveillance which is very robust, can be adjusted to signal noise and in the future perform other useful duties. Furthermore, surveillance cameras are reasonably priced, easy to install, and provide a sense of security to many people, especially elderly people that feel rather vulnerable. In the past a single camera system was useful, but would cause the subject of the system to be scaled depending on the camera view. Also, the subject has a higher chance of occlusion behind object like a couch or chair when a single camera is used. Therefore multiple cameras are implemented to know the exact size and become resistant to occlusion because of other views. Another advantage of the

multiple camera system is the exact point of space can be measured and therefore more accurate formula can be produced. The individual having an event can be classified by the velocities and angular motion because of accurate spatial representation. To accomplish data extraction the video object plane (VOP) is used to segment the person(s) falling. The speed components in the x, y, and z directions is also used and combined with temporal component t. Additionally, the video cameras are positioned in orthogonal directions so that they provide maximum coverage with minimal occlusions. In cases where more than one person is in the video assume the person fallen can get help from the other people. Thus action taken in the multiple person case is to announce through speakers a person has fallen. When the person is alone and a fall event is decided, the system will ask them if they are ok. An answer of yes, or if there is no answer after repeating the question many times, and each time waiting for a reasonable period (the number of times the question is asked, and the waiting time for an answer are parameters that can be adjusted during the installation of the system) will result in help event. A search in the database of the system to find the name(s) and number(s) of people to call for help will be executed. If the people in the support group do not respond then the system can call 911 providing the necessary information including the name and address of the person just fallen. To provide the services the system consist of several major components. Segmentation, people recognition (face recognition, size recognition, mannering, voice recognition, and habitual behavior), fall detection, client/server communication-network programming, database management, voice recognition, and administrative software. This thesis is structured as follows, background, network and administration, classification algorithm and implementation, followed by conclusion, references, and appendix.

Chapter 2: Background

In the area of fall detection there has been some research [1]-[45]. The devices used to gather data to determine a fall can be categorized into ambulatory devices and passive devices. Invasive devices will consist of devices that must be maintained by the user, while passive devices will continue to operate with minimal maintenance. Ambulatory devices in general consist of accelerometers, gyroscopes, magnetometers, goniometers, sole pressure sensors, pedometers, and actometers [34] . Static passive devices for monitoring falls consist of video cameras, microphones, floor sensors, IR sensors, and temporal contrast vision sensors. Then after the data has been gathered it must be classified to decide if an event has occurred. There have been many methods used to create a classification for falling including structured prediction, supervised learning, clustering, outlier detection, dimensionality reduction, and neural networks [1]-[45].

Classification using Naive Bayesian probability [10][45] has been employed to study data from video, floor vibrations and sound. In 2005 Cucchiara, et al [10] used computer vision to monitor a subject and classify the posture of the person. The research addresses the problems of shape change and occlusion by introducing two steps, blob classification and track-based classification. In step one a silhouette was extracted using the statistical and knowledge-based object detector which is based on background subtraction. In step two after the correct object has been determined a tracking module is employed to assist with occlusions. The Bayesian classifier that was created can exploit both the tracking information and the histogram associated with the silhouette. Another team Zigel, Litvak, and Gannot[45] used floor vibration and sound sensing to detect the data required to determine a fall. The idea of the team was to use an

accelerometer to monitor floor vibrations and a microphone to listen for abrupt loud events. The researchers used an energy calculation on the noises and vibrations which resulted in a normal statistical distribution for ambient events. Then a Bayes classification algorithm was used to determine a fall versus other events. Another Bayesian type classifier, relevance vector machine, was utilized by Jiang, et al [17] to classify postures obtained from a video after a fall is thought to have occurred. If the posture is "lie on floor" then the fall has been verified.

Support vector machines with training models have been used by a variety of researchers [30][43][23][9][24]. Qian, et al [30] the extracted body features by use of bounding boxes which are found based off the height of the blob and human body ratios. The features are used as input to three SVMs. The first SVM is used to decide if the subject is moving or stationary, SVM two is used to classify activities like walking and jogging while svm three is used to discover a fall versus sitting or squatting activities. Another team, Nadi, et al[24] also uses body features as input for their SVM, but once the features are extracted they form a Histograms of Oriented Gradient. This histogram will have encoded in it the orientation and motion that is occurring to the subject. Therefore, when the histogram is used as input for the SVM a fall will be detectable because of orientation and motion. While [30][24] is using features like the neck, chest, pelvis, or face, Yu, et al[43] uses ellipse fitting on a silhouette and then a projection histogram for different postures. The authors state the ellipse fitting is better than rectangle fitting because noise will not have as much affect. The ellipse being found is used to make postures which are classified by a SVM. Mirmahboub, et al [23] also uses silhouette's extracted from video, however instead of using body features or fitting the authors use a generalized silhouette. Claiming through observation there are certain aspects changed to the silhouette area during a

fall the authors state motion speed is represented. The silhouette area is considered the input to an SVM to classify falls. Cheng, Wen-Chang, and Ding-Mao Jhan [9] use triaxial accelerometers which are worn on the ankles , chest and waist. They compare all the areas and several classifiers ultimately deciding the chest location combined with a cascade-adaboost-svm gives the best fall detection. Bian, et al [7] use the Microsoft Kinect for body part tracking by use of a randomized decision tree. The 3-D body joints are tracked and a trajectory is made based off the joint distance which is used as the input to the SVM. This joint distance trajectory is the distance between the joint and the floor and includes the fall motion information.

Another tool used is the heuristic decision tree [2][5] to determine if a fall has just happened. Ariani, et al [2] uses data gathered from a wireless ambient sensor which uses both microwave and infrared for motion. The motion detector are set up in a zone fashion with overlapping zones. According to the researchers the result is a graph of each node that corresponds to cliques and each clique will be separated spatially to identify how many people are present. The fall detection is applied to each person as being a fall when only the lower half of the sensors are activated. Bai, Ying-Wen, Siao-Cian Wu, and Cheng-Lung Tsai [5] use the 3-axis accelerometer in a smart phone with the decision of a fall when four different states have happened. The states checked are all based all the motion of the accelerometer, including the weightlessness to impact the forward motion and if the motion had an overturn.

One class support vector machines(OCSVM) are also used by researchers [39][40][42] to find outliers in the data. Yu, M. et al [39] begins by using 3-d video features of the centroid and orientation of the person. The orientation is based on the voxel of the person and its calculated eigenvectors. In all Yu uses five features three of which are derived from the centroid and two

are from the orientation. The researchers note the SVM is the least computationally intensive one class classification technique they used, but performed worse than the single class minimax probability machine. Yu, et al [40] also uses 3-D video construction and the same features, however OCSVM is enhanced by using an optimized kernel function with negative training samples to separate the features further. Yu, et al [42] Leaves the 3-D scheme in favor of a one USB camera system to reduce economic cost and also introduces an online OCSVM learning algorithm. The researchers use ellipse fitting, shape structure, and position features to build a model of normal postures. Additionally, to reduce the number of non-falls considered a fall they introduce two rules which are the time of recovery to a sitting posture and the amount of motion detected from motion history image.

Fuzzy logic is another approach used to find an outlier event like falling used by [8][31][44] with the use of video. Brulin,Benezeth, and Courtial [8] first find all moving objects, which are tracked and classified as human or not with the Viola/Jones method. The human voxels are rectangle bounded and then have four features extracted from it. These features include the ratio of the distance from the center of gravity to the height, the width to the height. Next, principal component analysis performed on two different axis are used to recognize the posture and used as input for the fuzzy logic system. If a person is in the lying position for a set period a fall alert will be triggered. Zambanini, Sebastian, Jana Machajdik, and Martin Kampel [44] also use a bounding but in the form of an ellipse and is performed on a 3-D voxel created from two cameras. The orientation of the ellipse, axis ratio , and motion speed based on the relative number of new voxels are used to classify postures and confidence values in a fuzzy logic system for falls. Rezaee, Khosro, Javad Haddadnia, and Ahmad Delbari [31] also uses an

ellipse, however three measurements of motion are extracted. Motion history image is used to make the C motion coefficient, the speed of motion is found with the Euclidean norm, and the direction of motion is found and used as the inputs into the fuzzy logic system to determine abnormal walking which could indicate a fall.

The authors of the next paper Rougier et al [33] use a Gaussian mixture model to analyze human shape deformation. Video from one camera is used to obtain data points so the first problem is to extract the silhouette. The authors propose to use the canny edge detector in addition to a background subtraction method which takes into account shadows, highlights, and image compression. Shape analysis is the next step in the process with Procrustes analysis used to detect abnormal shapes. The full Procrustes distance is high in the case of a fall. However, to determine if a fall has occurred a Gaussian mixture is trained with normal activities and a fall will be discovered when there is an outlier detected. Additionally, the authors use multiple cameras that each are granted a vote for fall detection. Ultimately the additional cameras decrease the error rate of 10% for each camera to an error rate of 2.7% while having occlusions present.

Linear Discriminant Analysis used by Aziz, Robinovitch [4] and Manifold learning used by Hsu et al [16] both attempt to classify a fall by using only dimensional reduction. Aziz [4] gathered three-dimensional acceleration data by having eight video cameras with the participant searing a total of twenty-two reflective skin markers. Five anatomical positions are considered as important, the ankles waist, sternum, and top of the head. The mean acceleration and variance acceleration for each location based on the x, y, z are calculated and ran through the LDA. The falls and regular activities are each classified from training with results ranging from 54% to

98% based on location and quantity of markers used. Hsu [16] also uses video cameras, however it uses human silhouettes extracted with Gaussian mixture model which are used to find the centroid and bounding box. The researchers are studying open areas so they use nine different walking models to accommodate different view-angles and locality preserving projection. According to the authors the LPP approach is able to preserve the local structure and sample space. To detect abnormal activities the walking spaces are used in conjunction with the motion of the segmented features found by using the Hausdorff distance.

Hidden Markov models [37][17][1][36][19] are the next group of classifiers (HMM) employed to discover information about the data set extracted. Jiang [17] uses HMM in conjunction with RVM to detect falls. However, the researchers [1][37][36][19] use HMM as their primary classifier. Tong, et al [37] use a tri-axial accelerometer connected to the trunk of the person to determine the acceleration. The data collected from the accelerometer was used to train the HMM in order to describe the fall process before impact. This allows the prediction of falls before they occur by less than 400ms. On the other hand Jiang [17] uses video surveillance data for detection fall events. The study uses HMM to analyze the motion of the person, based on a new intensity of motion number, to predict if a fall is occurring and posture is analyzed by relevance vector machine to verify. HMM in this paper is setup to detect an acceleration phase, deceleration phase, and an inactivity phase. The HMM is configured using a multiple observation based training algorithm introduced by Xiaolin, Parizeau, Plamondon, and Rejean [17] giving weights to observation sequences. Anderson [1] obtains the silhouette from video that is a binary map showing body position. The method inputs the bounding box of the segmented individual into the HMM using the Baum-Welch procedure for each state(Falling,

kneeling, etc) with the likely state sequence found by using the Viterbi algorithm [1]. Researchers Thome, Miguet, and Ambellouis [36] use multiple cameras to create a multiview pose classification. The cameras each attempt to classify the pose by ellipse fitting which is then used to determine the posture of the subject. The multiple views are combined to increase the rate of detection for the current posture called the multiple view pose detector. The posture has now been discovered so motion is analyzed with a layered Hidden Markov Model which has input of the combined angle deviation between views. Finally, Khan, Zafar, and Won Sohn[19], one camera is used but the silhouette features extracted are ran through an R-transform accounting for scale, periodic, and invariant features making it robust to noisy data. The resulting information from the R-transform is the directional features with the most energy. The next step the researchers did was to limit difference in between data clusters and maximize it between the clusters themselves with kernel discriminant analysis. An HMM based on the KDA is used to classify different activities and recognize abnormal events such as a fall.

Another method to detect a fall is using a threshold number the researchers believe is tied to a fall having occurred [38][29][41][25][32][12][3]. Each introduced a cutoff measurement to detect a fall. Auvinet, et al [3] use a multiple camera network to reduce occlusion and create the 3-D shape of the person. The volume of the silhouette is determined, if the majority of the volume is occurring below 40 cm and has happened for a predefined amount of time then a fall is indicated. The researchers Yu, Miao, Naqvi, and Chambers [41] also use multiple cameras to enable head tracking of horizontal and vertical velocities. The velocities of the head are determined by calculating motion which is determined off of motion history image and tracking the head with a particle filter based on the condensation algorithm. If the velocities of the head

pass a certain point a fall will be determined as an event. The papers [29][38][25][32] all use one camera to gather data with Ozcan, et al [29] actually placing the camera on the subject. Rougier, et al[32] use motion history image and when motion has been found then the human shape is matched to an ellipse. If the motion is large and the ellipse has an orientation perpendicular or parallel to the camera with lack of motion a fall is indicated. The authors Nair, Rohit, and Bing [25] note shadows can affect the ellipse fitting of the subject so the remove them prior to fitting. The researchers Vaidehi, et al [38] claim motion is not needed to determine a fall and instead use only aspect ratio and inclination angle of the subject. The aspect ratio is the width of the person over the height while the inclination angle is compared to the camera angle. If the angle of inclination angle is between 45 and 90 with the aspect ratio larger than 1 a fall is signaled. Ozcan, et al[29] place a camera on the subject and the oriented image gradients and strengths for each edge are made. A fall is discovered if the change of the edge orientation is changing over each frame and the optical flow shows lots of motion. Estudillo-Valderrama, et al [12] use wireless accelerometers which send all the information to a personal server. These devices are placed on the human trunk to capture velocity and vertical angles of the subject. Additionally, knee falls are found because they estimate the curve approaches the normalized waveform.

The next classification type uses clustering to define event types and is used by [15][6]. Motion detection by using an asynchronous temporal contrast vision sensor is used by Fu, et al [15], with advantages being a built in anonymity, small bandwidth needs and the relative speed of data update. The authors propose the sensor can report a fall at ten times better speed than a regular video camera with practically instantaneous motion vector computation and fall event

reports. The amount of motion and speed of each event is clustered around a centroid velocity which allows the report of event type, having falls creating a large burst of pixels corresponding to velocity. Banerjee, et al[6] uses a Kinect sensor to extract the silhouette of and maintain the privacy of the subject. The silhouette is used to calculate the centroid based on the Zernike image descriptors. To classify each event the method used was the Gustafson-Kessel method[6].

The last classification type is the neural networks [18][26][21]. These studies all use cameras with Ma, et al [21] using an IR camera in the form of the Microsoft Kinect. Silhouette extraction by Juang, Chia-Feng, and Chia-Ming Chang [18] has a discrete Fourier transform performed on the vertical and horizontal histogram projection to eliminate size and position influence. In addition, to the discrete Fourier transform coefficients the authors use the length-width ratio of the silhouette to construct the neural network to classify the posture of the subject(standing, bending, sitting, lying). A fall is determined to have occurred if the time that the posture changes from standing to lying is less than 3.3 seconds and the lying posture has been for a certain number of frames. In [26] the authors Ngo, Nguyen, and Pham also use video, however instead of extracting only a silhouette they are concerned with fitting the silhouette with an ellipse in order to find features of the subject. The features of importance in the ellipse are the centroid, vertical angle, and major/minor axis. The authors use the data collected to train a two-layer feed-forward neural network claiming the results will be able to determine the direction of the fall. Another team, Foroughi et al, use the feed-forward neural network [14] deciding to use the motion as input. In order to capture the best motion index the authors used integrated time motion image techniques and applied a PCA method to get eighen-motion. The last paper by Ma et al [21], uses infrared camera in the form of the Microsoft Kinect to capture

motion and depth information. The authors extract the silhouette information with the depth camera and avoid color images to maintain privacy for the subject. The curvature scale of space is extracted for each frame because it is invariant to translation, rotation, scaling, and local shape deformation [21]. The neural network which is a single-hidden-layer feedforward neural network using particle swarm optimization, is trained for fall detection.

Chapter 3: Network and Administration

I. TCP vs UDP

The Internet Protocol(IP) supports two transport layer protocols either Transmission Control Protocol(TCP) or User Datagram Protocol(UDP). When making a decision about which protocol to use the programmer should consider the needs of the application. In this paper the application consist of a multithreaded server/client paradigm and the continuous lossless transfer of video. Determining factors to consider are the speed of transfer, the scalability of clients, and the reliability of data delivery.

Speed of data transfer is the most important factor when considering which protocol to use with regards to video. The reason the data transfer rate is so important is because of the sheer size of the video data being transferred. A given example can be see with a RGB video frame at 800 X 600 pixels. The size of the frame in BMP format will be 480000 pixels, with each pixel using three bytes for twenty-four bit true color giving a total size of 1440k. Furthermore, the video frame rate is usually thirty frames per second which means the total data transfer rate for one client every second will be 43.2Mbytes. The concept remains true if video compression is used even though it is starting with smaller numbers like 144k in jpeg format for a video frame, which results in each client needing 4.32 Mbytes. In a situation where the network is optimal, solely being used for the application, and each room requiring a minimum of two cameras the rate data transfer being of high importance becomes apparent. The question becomes which of the two protocols is better at high speed data transfer, TCP, or UDP. The answer is UDP. UDP is far faster than TCP for the following reasons; it has no error-checking, no reliability control, no congestion control, and no flow control.

The types of errors that can occur in a network transfer include the loss of packets, the duplication of packets, the order of packets received being incorrect, and the corruption of a data packets. UDP only accounts for data corruption at the packet level, which is basically a checksum like that used in TCP. TCP will also check for the other type of network errors, which results in increased network traffic and slower data transfer speed. The way TCP accounts for errors that deal with packet order, loss of packets, and packet duplication relies on the three way handshake. During the handshake the sender will create a sequence number and send it to the receiver. The receiver will return to the sender its own sequence number and an acknowledgment number to indicate a start of a session and receipt of the sender's packet. The final step of the three way handshake is to acknowledge from the server to the client receipt of the last packet. The connection is established, so the acknowledge and sequence process continues until the payload is finished. If during the transfer process a sequence number is not received after a certain amount of time and no acknowledgement is sent then the client will resend the packet. Additionally, if the server discovers the checksum is incorrect it will throw out the data and wait for the resend. As a result, the data is guaranteed to arrive in order due to the sequence number, not be corrupt because of the checksum number, and not have any missing packets because an acknowledgment will not be sent. However, the majority of the packets have been sent without problems, meaning there is excessive traffic from acknowledgments during the TCP protocol use. Furthermore, TCP also has flow control and congestion control which tend to throttle back data transfer rates too much. For example flow control will occasionally cause the receiver to tell the sender it is not ready for any information because it cannot handle any more. The result is the sender will attempt to recover the session by starting with a small packet having

the idea of increasing it again over time. The small packet will be inefficient though because it will send a very small amount of data to the relative size of the TCP header. Similarly the congestion control will also slow the data transfer rate too far and can occur when the packets have been lost or timers show acknowledgments are taking too long.

Scalability of clients is next feature in the system that is important because each area must have a minimum of two cameras. In an area the number of clients can climb very high for a large square footage building. In this case the network data transfer rates are the main limiting factor for adding new clients. The server is using a multi-threaded approach and is essentially limited by memory. UDP is the clear winner for scaling up to a higher number of clients for data transfer speed reasons mentioned before. However, an example with some numbers can make it more clear for the reader. Given the previous data transfer numbers of 4.32Mbytes each second per client for raw data transfer then the additional data overhead of the ack packets is about two percent. The calculation is found by taking the maximum transfer until of 1500 bytes and dividing them into the 4.32Mbytes giving a total of 2880 packets sent each second. These packets in TCP format will each have a header of 20 bytes while the UDP header will only require 8 bytes. This means TCP headers alone will require an additional 34560 bytes or .08% of the data. The use of TCP will also require roughly an addition 2880 packets in the form of acknowledgment packets with each packet requiring 20 bytes for total of 57600 bytes. Combined with the previous TCP overhead and the additional data needed per second is 92.16k or a two percent overhead in data alone. Recall the speed of the data transfer will also be affected by the TCP window from the flow control and the congestion control throttling which often is larger than needed.

Reliability of data delivery is also important in the application because each video frame is used to determine if a fall has occurred by use of the motion captured. If one or multiple frames are missing, then the velocity of the silhouette will be considered much higher than it actuality. A fall may be erroneously interpreted if the velocity calculations are inaccurate. In this situation TCP is the superior protocol because it already has a built in way of handling data and is guaranteed to be received by use of the acknowledgment packet. However, UDP does have a guarantee of data integrity in each datagram and also an option to use a checksum for the entire payload. TCP also guarantees the frames will arrive in correct order without errors as discussed previously.

The end result of protocol choice falls to UDP because of the data transfer speeds being superior to TCP. There are some considerations to take into account; the use of UDP means the application must handle errors that can occur from a network transfer. Network errors include the loss of packets, the duplication of packets, the order of packets received being incorrect, and the corruption of data packets. The main advantage of UDP is that network errors can be handled in a more efficient manner than TCP because the majority of the packets will arrive in the correct way. This means the number of acknowledgment packets can be significantly reduced.

II. Multithreaded Server & Client

When deciding the network paradigm of server and client the programmer has a couple options such as asynchronous/synchronous, multithreading, and port sharing use.

Deciding to use an asynchronous server over a synchronous server allows the clients to send datagrams at maximum transfer because the server will not need to wait for another process

to finish. In the scheme chosen speed is very important and the video must be complete or not missing any frames. Again, these factors are important because time must be consistent in the data calculations. However, datagrams can arrive out of order and be lost. The server must be able to handle each client independent of the others, requesting data when needed and adding to the video buffer as frames arrive. Additionally, an asynchronous server is the perfect type for multithreading allowing all data to be handled at arrival. As a result, cross threading issues can arise and should be handled such that data is consistent and reliable. This means that all of the errors that TCP handles must be handled by the server and client. Packets must be in correct order, lost packets must be requested and duplicate packets must be ignored properly. To have appropriate data manipulation time a buffer is implemented on the server for each client. A popular scheme for UDP server reliability is to implement TCP reliability in the application layer. Generally this means the clients must be able to have flow control adjusting their speed up or down. To accomplish flow control the client will often adjust just like TCP, when acknowledgment packets are slow or missing. Accordingly, the server must send acknowledgment packets, which is the main way reliability can be achieved while still being much faster than TCP. The server accomplishes the speed optimization by limiting the number of acknowledgments, perhaps to one for every five packets. Last to consider is the port sharing for each video client to have its own port. The reason for this is because each port will always be available and not need to be handed off before serving the client.

Administration

I. Server

The server administration in the sense of this project is not dealing with the operating system or physical maintenance, instead it is referring to the jobs of the software server. The jobs for the server include the processing of frame data, classification, subject status on alarm, and phone alert. When the server receives raw data from the client, the video must be pre-processed for video object plane and then the relevant object must have important features extracted. These features are the inputs to the classification algorithm, which in this paper decides if a fall has happened to the subject. When a fall has been determined, it must be verified through voice questions. After verification an alarm will be sent by telephone to the support group in the database. Consequently, the database must also be updated and maintained by the server. To maintain the database information as current, the server will occasionally call the support group. During the call the server will use voice recognition to ask if the person is still a support individual, if not then a notification to the owner will be given.

II. Client

The client must capture video and then transfer it to the server. Therefore a video camera, and some type of network must be available to the client. Cameras available have different resolutions as well as viewing angles. In the setup used in this project the camera had up to and available 1920x1080 resolution with a diagonal viewing angle of 128 degrees provided by a two megapixel CMOS sensor. The camera also had both ethernet and wireless support with different video compression algorithms. Additional setup required a total of two cameras to be setup perpendicular to each other at roughly the same height to provide depth in the constructed view. Three dimensional space allows calculations to ignore camera perspective. An example of perspective changing is when a subject is closer to the camera they will appear larger. The result

is the calculations will be changed because the distance traveled will appear different, hence the second camera is necessary without additional preprocessing.

III. Security

Security is quite important in this project because ultimately the system could be in places where privacy is important, say the restroom. This means the system must be able to provide anonymity of people from visual predators and events that could be used against them. To assist in privacy the server and the clients will be on an internal network that is not connected to another network. Another precaution that can be used is encryption of the video frame sent from the client. If encryption is implemented then a wireless network should be meet any privacy needs. Also, the server is implementing a system that can make a request to a subject if help is required so it will not be necessary for third party verification. This means visual confirmation from a third party in the form of video is unnecessary and the only place video will be available is on a server that is not connected to an external network. While the current system is sending the entire video feed, it is possible to change the implementation to only send the silhouette of the person which will preserve features.

IV. Telephone

The telephone communication is used in the case of an event and is used to alert the proper authorities for assistance. The system can make a call directly to 911 or support service with the proper information for the person in trouble. Name, age, weight, and any other attributes that may be useful can be relayed. The system has several choices for making emergency calls such as, a dedicated landline, cell phone, or VOIP service. Each different service has benefits and disadvantages. Land lines are very good at robust service. However, if

the server is in a room that does not already have a line the cost to run one may be high. Cellular phones could allow the server to be placed almost anywhere, but the signal strength of the phone could be an issue. Another problem with cell phones is the providers sometimes lock the phone, which could have unforeseen consequences. In this paper a cell phone was implemented via the computer acting as a bluetooth headset which requires the proper drivers and operating system. The last option, voice over IP, has many services available but would require a connection to the internet. The connection could be an issue if the server is not setup correctly. Also, if the emergency service is a direct connect to 911 then most VOIP services cannot provide the ability to the user.

V. Voice Recognition

The voice recognition system consist of a speaker, microphone and software. Some of the issues that come up are wrong word recognition, speaker echo, and microphone sensitivity. To combat wrong word recognition the system is designed in such a way that only simple yes or no responses are requested. This idea has the added benefit that an injured subject will not be required to have a fully conscious understanding of their situation. Speaker echo is another problem that occur, which is when the microphone will react to the speaker. To solve speaker echo the recognition system is muted during status inquiring phase. Microphone sensitivity is another problem to consider. Ambient noise on a microphone that is too high and microphone sensitivity set too low can cause recognition problems. The microphone should be placed and adjusted according to room acoustics with a possible voice filter for the recognition system.

Chapter 4: Fall Detection Algorithm and Classification

The fall classification algorithm is part of a larger system which includes face recognition, performs distance resolution based on camera super triangulation and is capable of following one or more people. In this research paper the gray scale channel of each video frame is used to segment and extract the video object plane of the person. The (x,y,z) coordinates of each of the pixels in the video object plane are computed. Additionally, the center of gravity (x_c, y_c, z_c) of the video object plane is calculated. When a person is falling or sitting the velocity vector of the center of gravity has a negative z component. The difference between falling and sitting is the velocity of the center of gravity during fall is usually greater than the velocity when sitting. Denoting by **V(**x,y,z,t) the video object plane of the person of interest then:

Equation 1: Velocity of VOP

$$\frac{dV(x,y,z,t)}{dt} = \frac{\partial V}{\partial x}\frac{dx}{dt} + \frac{V}{\partial y}\frac{dy}{dt} + \frac{\partial V}{\partial z}\frac{dz}{dt} + \frac{\partial V}{\partial t} \qquad (1)$$

Where $\frac{\partial V}{\partial x}$, $\frac{\partial V}{\partial y}$, $\frac{\partial V}{\partial z}$ is the rate of change for the projections of the video object plane on the x, y, z axis. Thus for every frame estimate the projections of the video object plane on the (x,y,z) axis, then obtain the difference of the projections from one frame to the next. The differences are an estimate of the $\frac{\partial V}{\partial x}$, $\frac{\partial V}{\partial y}$, $\frac{\partial V}{\partial z}$. The ($\frac{dx}{dt}$, $\frac{dy}{dt}$, $\frac{dz}{dt}$)=(v_x, v_y, v_z) is estimated by subtracting the center of gravity of the current video object plane from the center of gravity of the previous video object plane of the person being tracking. The $\frac{\partial V}{\partial t}$ is estimated as the time, which can be

measured in the number of frames or the time. The measured velocity begins from the fall start until the person hits the floor or ground. The person is determined to have hit the ground when more than the feet of the person are in contact with the ground, and the $\frac{\partial V}{\partial z}$ component is close to zero. Therefore, each fall or sitting on the floor action begets an estimate of these seven components which is denoted from now on as:

Equation 2: Rate of Change

$$x_1 = \frac{\partial V}{\partial x},\; x_2 = \frac{\partial V}{\partial y},\; x_3 = \frac{\partial V}{\partial z},\; x_4 = \frac{dx}{dt}\; x_5 = \frac{dy}{dt},\; x_6 = \frac{dz}{dt},\; x_7 = \frac{\partial V}{\partial t} \quad (2)$$

Let N be the number of videos simulating fall activity. For each of the N videos an estimate of the above seven parameters $(x_{n1}, x_{n2}, x_{n3}, x_{n4}, x_{n5}, x_{n6}, x_{n7})$, n=1,2,3,…,N represent the seven parameters obtained by the nth video of simulated fall. Next the computation of the centroid for the fall class is the average of the N vectors. Thus:

Equation 3: Centroid of Falling Class

$$\overline{X}_f = \frac{1}{N}\left(\sum_{n=1}^{N} x_{n1}, \sum_{n=1}^{N} x_{n2}, \sum_{n=1}^{N} x_{n3}, \sum_{n=1}^{N} x_{n4}, \sum_{n=1}^{N} x_{n5}, \sum_{n=1}^{N} x_{n6}, \sum_{n=1}^{N} x_{n7}\right) = (\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4, \overline{x}_5, \overline{x}_6, \overline{x}_7) \quad (3)$$

Also the estimate of the variance covariance matrix from N video object planes of the N video data simulating fall is computed.

Equation 4: Variance/Covariance Fall Matrix

$$\sum_f = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} & \sigma_{17} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} & \sigma_{26} & \sigma_{27} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} & \sigma_{35} & \sigma_{36} & \sigma_{37} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} & \sigma_{45} & \sigma_{46} & \sigma_{47} \\ \sigma_{51} & \sigma_{52} & \sigma_{53} & \sigma_{54} & \sigma_{55} & \sigma_{56} & \sigma_{57} \\ \sigma_{61} & \sigma_{62} & \sigma_{63} & \sigma_{64} & \sigma_{65} & \sigma_{66} & \sigma_{67} \\ \sigma_{71} & \sigma_{72} & \sigma_{73} & \sigma_{74} & \sigma_{75} & \sigma_{76} & \sigma_{77} \end{bmatrix} \quad (4)$$

Let M be the number of videos simulating sitting down, lying down, and other activities related to reaching the floor or the ground. For each one of these M videos the estimated seven parameters $(x_{m1}, x_{m2}, x_{m3}, x_{m4}, x_{m5}, x_{m6}, x_{m7})$, m=1,2,3,…,M represents the seven parameters obtained by the mth video of simulating reaching the floor or ground intentionally. Next, compute the centroid of the non-falling class as the average of the M vectors. Thus:

Equation 5: Centroid of Non-Falling Class

$$\overline{X}_g = \frac{1}{M}\left(\sum_{m=1}^{M} x_{m1}, \sum_{m=1}^{M} x_{m2}, \sum_{m=1}^{M} x_{m3}, \sum_{m=1}^{M} x_{m4}, \sum_{m=1}^{M} x_{m5}, \sum_{m=1}^{M} x_{m6}, \sum_{m=1}^{M} x_{m7}\right) = (\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4, \overline{x}_5, \overline{x}_6, \overline{x}_7) \quad (5)$$

Also, compute an estimate of the variance covariance matrix from M video object planes of the M video data simulating reaching the floor or the ground intentionally.

Equation 6: Variance/Covariance Non-Fall Matrix

$$\sum_g = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} & \sigma_{17} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} & \sigma_{26} & \sigma_{27} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} & \sigma_{35} & \sigma_{36} & \sigma_{37} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} & \sigma_{45} & \sigma_{46} & \sigma_{47} \\ \sigma_{51} & \sigma_{52} & \sigma_{53} & \sigma_{54} & \sigma_{55} & \sigma_{56} & \sigma_{57} \\ \sigma_{61} & \sigma_{62} & \sigma_{63} & \sigma_{64} & \sigma_{65} & \sigma_{66} & \sigma_{67} \\ \sigma_{71} & \sigma_{72} & \sigma_{73} & \sigma_{74} & \sigma_{75} & \sigma_{76} & \sigma_{77} \end{bmatrix} \quad (6)$$

Now, let $Y_f = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$, be a vector from a video with an unknown event where there is a fall or non-fall then the Mahalanobis distance of that vector from the centroid $\overline{X}_f$ is:

Equation 7: Mahalanobis Distance for Ground

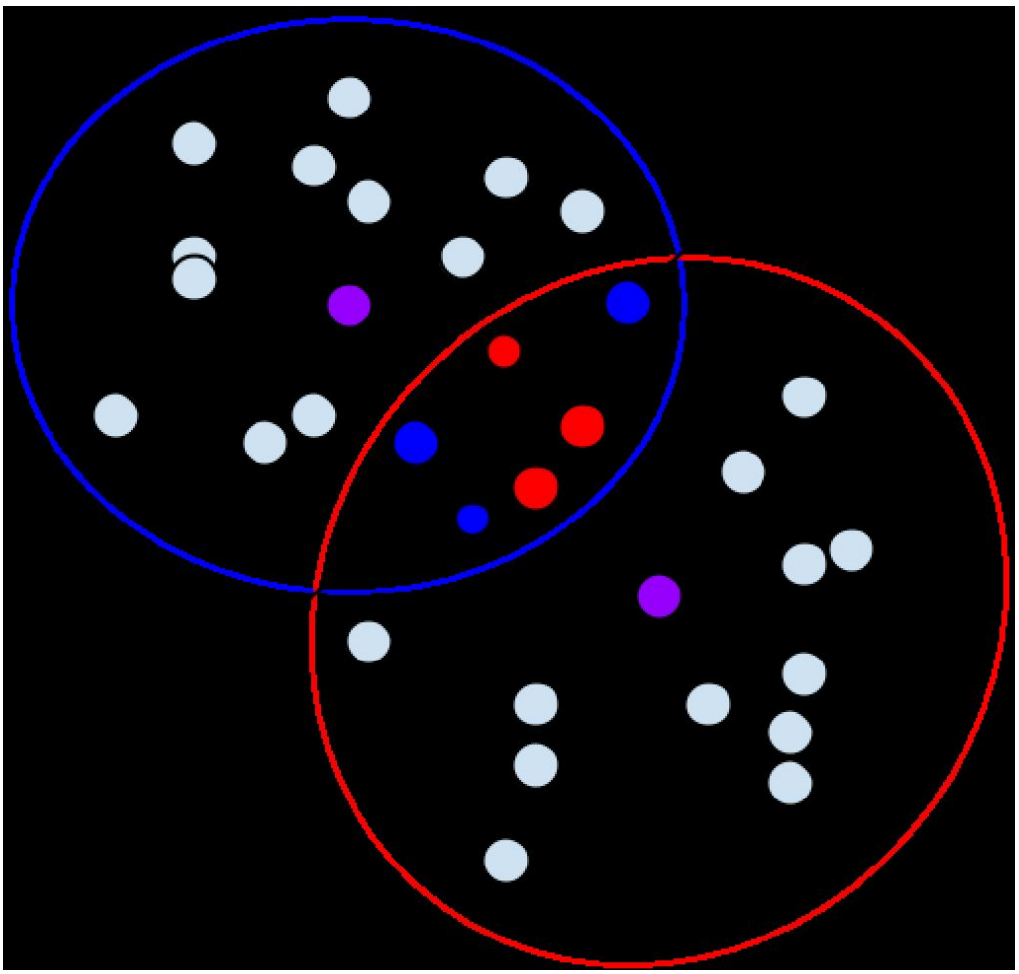
$$D_f = (Y_f - \overline{X}_f)' \Sigma_f^{-1} (Y_f - \overline{X}_f) \quad (7)$$

Also let $Y_g = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$, be a vector from an unknown event video then the Mahalanobis distance of that vector from the centroid $\overline{X}_g$ is:

Equation 8: Mahalanobis Distance for Fall

$$D_g = (Y_g - \overline{X}_g)' \Sigma_g^{-1} (Y_g - \overline{X}_g) \quad (8)$$

Figure 1. Classification by Intersecting Hyperspaces



Intersecting hyper spaces. The number of elements in the intersection that belong to space1 (blue circle) but have smaller distance from the centroid of space2 (red circle) than the centroid of space1 are misclassified. Also the number of elements in the intersection that belong to space2 (red circle) but have smaller distance from the centroid of space1 (blue circle) than the centroid of space2 are misclassified.

Based on the given data compute the centroid of each one of the two spaces as mentioned above. The hyperspace (seven dimensional space) derived from the fall data will be referred to as space1, and the hyperspace derived from the non-fall data as space2. If the hyperellipse of the one space includes data of the other space then the two spaces intersect, otherwise the two spaces have no intersection. When the two spaces are not intersected then the probability for correct classification is one (100%), and therefore the probability of misclassification is zero. Incorporating new elements in the classification vector which are not a linear combination of elements already in the vector, and are cross correlated with the output (have cross correlation significantly different than zero), increases the classification power and decreases the misclassification error. When the two spaces are intersecting then a misclassification occurs when a vector belongs to one space and the distance from the centroid of the other space is smaller than the distance of the vector from the space centroid it belongs to. An estimate of the misclassification error in this case is computed by taking all the cases in the intersection of the two spaces for which although they belong to one space or class their Mahalanobis distance from the centroid of the other class is smaller than the Mahalanobis distance from the centroid they actually belong, and divide them by the total number of points. Thus if N are the number of classification vectors from class 1, M are the number of classification vectors from class 2, k are the number of vectors in the intersection of the two hyperellipsoids of the two classes, and r are the vectors in the intersection having distances from their own centroid larger than the distance from the centroid of the class that they do not belong to, then the misclassification error estimate is:

Equation 9: Misclassification Error

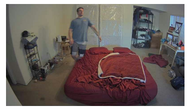
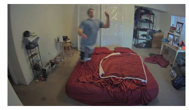
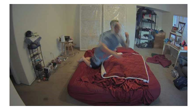
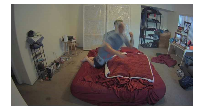
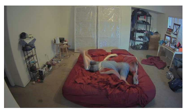
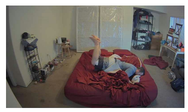
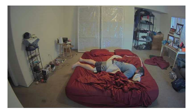
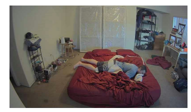
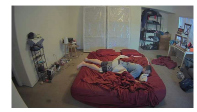
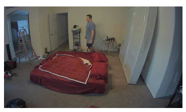
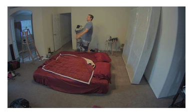
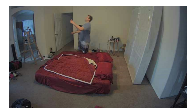
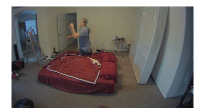
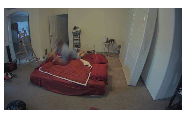
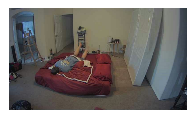
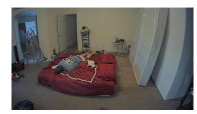
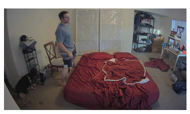
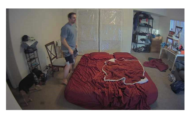
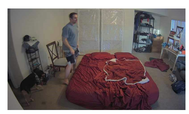
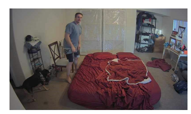
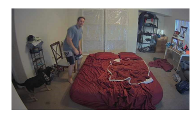
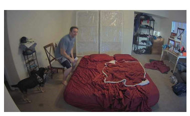
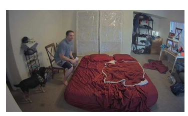
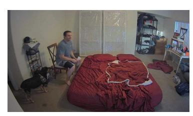
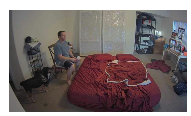
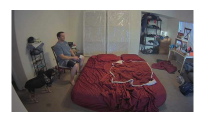
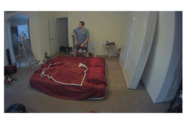
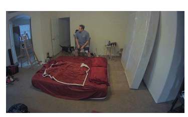
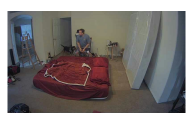
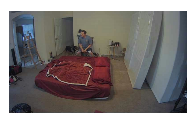
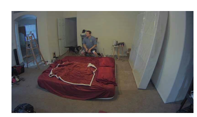
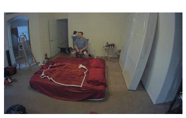
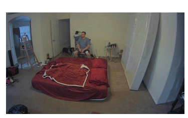
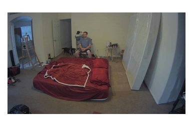
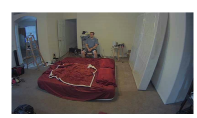
$$\text{Probability of misclassification error} = \frac{r}{N*M} \qquad (9)$$

Points not belonging in the intersection of the two hyperellipsoids are always being classified correctly and their probability of misclassification is zero. Thus if the two spaces do not intersect the probability of correct classification is one, and therefore the probability of misclassification is zero.

Choice of the Classification Vector Components

The (x,y,z) coordinate system is right handed with the z-axis vertical to the floor or the ground, and the (x,y) plane parallel to the ground. The vision system includes at least two cameras one facing the (x,z) plane and the other the (y,z) plane. The cameras are calibrated so that the triangulation software can resolve distances and also coordinates of points in the space that are in the view of both cameras. The cameras are part of a WiFi local area network and they transmit wireless using the 801.11 wireless protocol to a server where all the processing takes place including the classification algorithm when it is triggered. The choice of the inputs in the classification vector is based on a pre sampling method. Start with a small number of inputs that influence the output, and also test the amount of intersection of the two classes. As more relevant inputs are added then the separability of the two spaces increases. Based on pre sampling it is concluded that the seven components of the classification vector produce separate spaces, therefore classifying with probability of 100% and zero misclassification error.

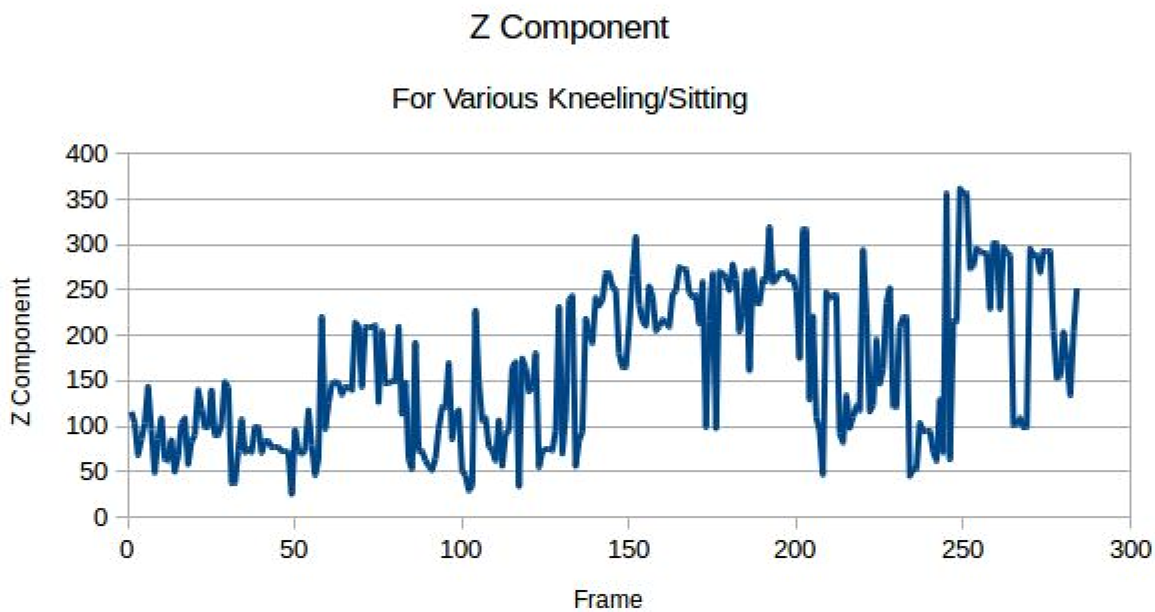
Figure 2. Video Cameras Orthogonal Setup View Results

Camera One View Trip Fall					
					
Camera Two View Trip Fall					
					
Camera One View Sitting					
					
Camera Two View Sitting					
					

Two cameras are capturing a fall and a sit from orthogonal positions. The video will have the silhouette extracted by using edge detection, combined with Gaussian Blur. Additionally, the image is made binary.

Based on the classification vector, the data obtained from the videos, the classification distance, the training of the classifier, the two derived classes have no intersection therefore vectors extracted from the falling videos are mapped to the fall class, and vectors obtained from the non-falling videos are mapped to the non-fall class. The reason for that is that when a person is sitting the ($\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z}$) which represents the rate of change of the projection of the video object plane of the person in the x, y, z, directions respectively does not change that much from one frame to the next, and the absolute value of the change is uniform across the three axis.

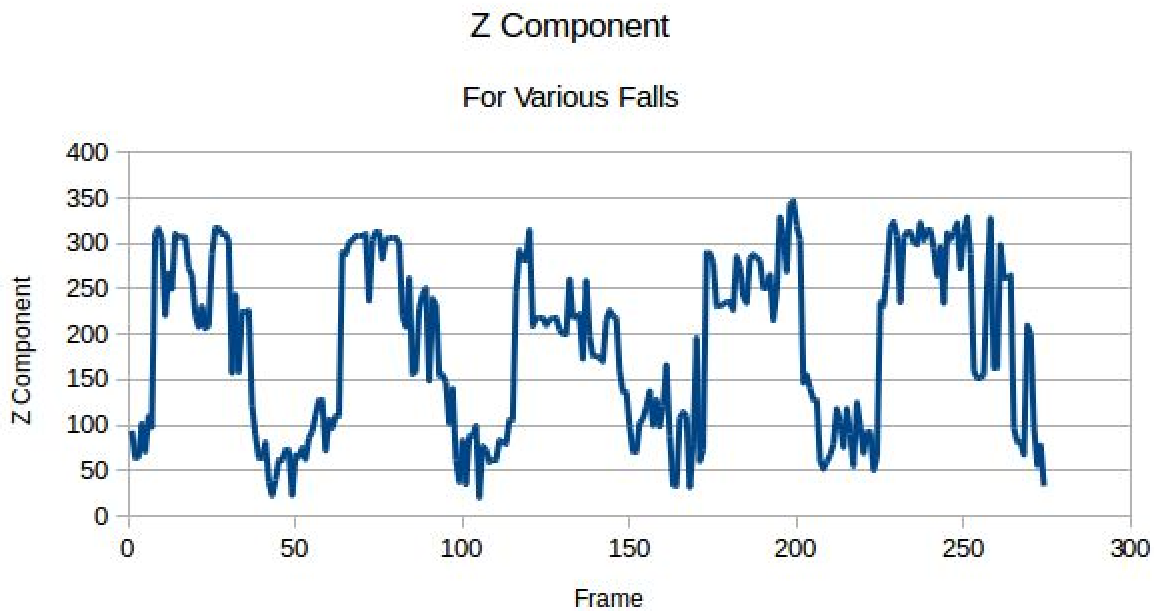
Figure 3. Graph of Z Projection Changes During Various Non-fall Events



Graph depicting Z projection changes during chosen non-fall events, non-fall events in Z are relatively smooth compared to fall events.

When a person is falling ($\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z}$) it changes much faster from one frame to the next, and the absolute value of the rate of change across the three axis is non uniform, meaning that in some direction(s) the projection(s) are much bigger than some other direction(s).

Figure 4. Graph of Z Projection Changes During Various Fall Events



Graph depicting Z projection changes during chosen fall events, fall events in Z are seen at each sharp peak.

An example of falls can be seen in the figure four graph where a fall is occurring at every sharp peak. The sharp decline is standing, and the middle peaks are getting up from the floor, movements onto the hands and knees and then into a squat. Also the velocity vector ($\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}$) =(v_x, v_y, v_z) of the center of gravity of the video object plane changes relatively slow in the case of sitting and much faster in the case of falling in addition to that the rate of change of the velocity vector in the case of sitting is uniform in absolute value, where in the case of falling is non uniform . Also the $\frac{\partial V}{\partial t}$ is relatively small in the case of sitting since it takes a relatively large

number of frames and therefore relatively large time from the time the action was initiated to the time that it was concluded, compared to the less number of frames and less time in the case of falling. These are the reasons that the centroids and the spaces defined by the vectors of the data extracted by the sitting videos are well separated from those extracted by the falling videos.

Chapter 6: Conclusion

The older section of the population is continually increasing every year and falls are very common for them. Often a fall will have severe health implications including fractured bones, and others that if left unattended could result in premature death. Occasionally people pretend to fall for the purpose of insurance fraud to legitimate business owners. Cameras are very common, especially surveillance cameras and cell phone cameras, recording just about all activities. Classification algorithms based on video have been constructed in order to detect falls. In this research paper there is a constructed algorithm based on frame segmentation, and speed components in the x, y, z directions over time t. The speed components are estimated from video obtained by cameras positioned in orthogonal directions which provides views from each angle. This algorithm separates a fall from other activities such as sitting on the floor, or lying on the floor.

Appendix: Code Excerpts

i. TCP Asynchronous Server Selected Excerpt

```
public void StartListening()
    {
      // Bind the socket to the local endpoint and listen for incoming connections.
      try
      {
        localEndPoint = new IPEndPoint(IPAddress.Any, port);
        // Create a TCP/IP socket.
        listener = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
              ProtocolType.Tcp);
        // listener.SetSocketOption(SocketOptionLevel.Socket,
              SocketOptionName.ReuseAddress, true);
        listener.Bind(localEndPoint);
        listener.Listen(1000);
        currentClient.workSocket = listener;
        while (true)
        {
          // Set the event to nonsignaled state.
          allDone.Reset();
          // Start an asynchronous socket to listen for connections.
          listener.BeginAccept(new AsyncCallback(AcceptCallback), listener);
          // Wait until a connection is made before continuing.
          allDone.WaitOne();
        }
      }
      catch (Exception e)
      {
        Console.WriteLine(e.ToString());
      }
    }

public static void AcceptCallback(IAsyncResult ar)
    {
      // Signal the main thread to continue.
      allDone.Set();
     // Get the socket that handles the client request.
```

```
        Socket listener = (Socket)ar.AsyncState;
        Socket handler = listener.EndAccept(ar);
        // Create the state object.
        clientObject state = new clientObject();
        state.workSocket = handler;
        Console.WriteLine("\tConnection from : {0}", handler.LocalEndPoint.ToString());
        handler.BeginReceive(state.buffer, 0, clientObject.BufferSize, 0, new
                AsyncCallback(ReadCallback), state);
    }
```

ii. TCP Client Selected Excerpt

```
    private static void Send(Socket client, byte[] data) {
        ushort size = (ushort)data.Length;
        byte maxSeqByte0 = (byte)size;
        byte maxSeqByte1 = (byte)(size >> 8);
        byte[] sendArray = new byte[data.Length + 3];
        sendArray[0] = clientNumber;
        sendArray[1] = maxSeqByte1;
        sendArray[2] = maxSeqByte0;
        Buffer.BlockCopy(data, 0, sendArray, 3, data.Length);
        size = maxSeqByte1;
        size <<= 8;
        size += maxSeqByte0;
        int count = 0;
        for (int i = 0; i < data.Length; i++) { count++; }
        client.BeginSend(sendArray, 0, sendArray.Length, 0, new
                AsyncCallback(SendCallback), client);

    }
```

iii. UDP Asynchronous Server Excerpt

```
 void startServer()
    {
        //listen on host and port
        //new video connection on incoming connection, video connection is thread of incoming
        IPEndPoint iep = new IPEndPoint(IPAddress.Any, port);
        server = new UdpClient();
        server.ExclusiveAddressUse = false;
        server.Client.SetSocketOption(SocketOptionLevel.Socket,
```

```
                SocketOptionName.ReuseAddress, true);
            server.Client.Bind(iep);
            portIp = iep.ToString();
            clientIPE = new IPEndPoint(IPAddress.Any, 0);
            status = "Starting Server";
          //Start listening.
           Thread listenThread = new Thread(new ThreadStart(Listening));
           listenThread.Start();
           status = "Listening";
           serverRunning = true;
        }

 private void Listening()
        {
            byte[] data=null;
            //Listening loop.
            while (true)
            {
                status = "Listening";
                //receieve a message form a client.
                data = server.Receive(ref clientIPE);
                int whichClient=0;
                whichClient = data[0]; //client number first byte

                if (whichClient < clientMax)
                {
                    clients[whichClient].video.addFrame(data);
                }
            }
        }
```

iv. UDP Client Selected Excerpt

```
        private void preprocessData(byte[] data)
        {
            //1472 bytes to work with -- mtu at 1500 - 20 bytes for ip header and 8 bytes udp header
            //bytes     | Data
            //0         | client
            //1-2       | Seq of frame
            //3         | frame
```

```
    //4 - 5
    //6-1471      | current frame data
     if (frame[currentFrame] == null)
    {
        framePacket newFrame = new framePacket(data);
        frame[currentFrame] = newFrame;
    }
    else
    {
        frame[currentFrame].updateFrame(data);
    }
    for (ushort i = 0; i < frame[currentFrame].maxSeq; i++)
    {
        sendInfo(frame[currentFrame].returnPacket(clientNumber,currentFrame,i));
        if (i % burstSpeed == 0) { Thread.Sleep(syncSpeed);
    }
    if (currentFrame != 254) { currentFrame++; }
    else { currentFrame = 0;  }
    //each packet start with client, seq, frame
}
```

References

[1] Anderson, Derek et al. "Recognizing Falls From Silhouettes." *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (2006): n. pag. Web.

[2] Ariani, A. et al. "Simulated Unobtrusive Falls Detection With Multiple Persons." *IEEE Transactions on Biomedical Engineering IEEE Trans. Biomed. Eng.* 59.11 (2012): 3185–3196. Web.

[3] Auvinet, Edouard et al. "Fall Detection With Multiple Cameras: An Occlusion-Resistant Method Based On 3-D Silhouette Vertical Distribution." *IEEE Transactions on Information Technology in Biomedicine IEEE Trans. Inform. Technol. Biomed.* 15.2 (2011): 290–300. Web.

[4] Aziz, O., and S. N. Robinovitch. "An Analysis Of the Accuracy of Wearable Sensors for Classifying the Causes of Falls in Humans." *IEEE Trans. Neural Syst. Rehabil. Eng. IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.6 (2011): 670–676. Web.

[5] Bai, Ying-Wen, Siao-Cian Wu, and Cheng-Lung Tsai. "Design And Implementation of a Fall Monitor System by Using a 3-Axis Accelerometer in a Smart Phone." *IEEE Transactions on Consumer Electronics IEEE Trans. Consumer Electron.* 58.4 (2012): 1269–1275. Web.

[6] Banerjee, Tanvi et al. "Day Or Night Activity Recognition From Video Using Fuzzy Clustering Techniques." *IEEE Trans. Fuzzy Syst. IEEE Transactions on Fuzzy Systems* 22.3 (2014): 483–493. Web.

[7] Bian, Zhen-Peng et al. "Fall Detection Based On Body Part Tracking Using a Depth Camera." *IEEE Journal of Biomedical and Health Informatics IEEE J. Biomed. Health Inform.* 19.2 (2015): 430–439. Web.

[8] Brulin, D., Y. Benezeth, and E. Courtial. "Posture Recognition Based On Fuzzy Logic for Home Monitoring of the Elderly." *IEEE Transactions on Information Technology in Biomedicine IEEE Trans. Inform. Technol. Biomed.* 16.5 (2012): 974–982. Web.

[9] Cheng, Wen-Chang, and Ding-Mao Jhan. "Triaxial Accelerometer-Based Fall Detection Method Using a Self-Constructing Cascade-AdaBoost-SVM Classifier." *IEEE Journal of Biomedical and Health Informatics IEEE J. Biomed. Health Inform.* 17.2 (2013): 411–419. Web.

[10] Cucchiara, R. et al. "Probabilistic Posture Classification For Human-Behavior Analysis." *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans IEEE Trans. Syst., Man, Cybern. A* 35.1 (2005): 42–54. Web.

[11] Doukas, Charalampos N, and Ilias Maglogiannis. "Emergency Fall Incidents Detection In Assisted Living Environments Utilizing Motion, Sound, and Visual Perceptual Components." *IEEE Transactions on Information Technology in Biomedicine IEEE Trans. Inform. Technol. Biomed.* 15.2 (2011): 277–289. Web.

[12] Estudillo-Valderrama, M.a. et al. "Design And Implementation of a Distributed Fall Detection System;Personal Server." *IEEE Transactions on Information Technology in Biomedicine IEEE Trans. Inform. Technol. Biomed.* 13.6 (2009): 874–881. Web.

[13] Fleck, S., and W. Strasser. "Smart Camera Based Monitoring System And Its Application to Assisted Living." *Proceedings of the IEEE Proc. IEEE* 96.10 (2008): 1698–1714. Web.

[14] Foroughi, Homa et al. "An Eigenspace-Based Approach for Human Fall Detection Using Integrated Time Motion Image and Neural Network." *2008 9th International Conference on Signal Processing* (2008): n. pag. Web.

[15] Fu, Zhengming et al. "An Address-Event Fall Detector For Assisted Living Applications." *IEEE Trans. Biomed. Circuits Syst. IEEE Transactions on Biomedical Circuits and Systems* 2.2 (2008): 88–96. Web.

[16] Hsu, Sheng-Bin et al. "Falling And Slipping Detection for Pedestrians Using a Manifold Learning Approach." *2013 International Conference on Machine Learning and Cybernetics* (2013): n. pag. Web.

[17] Jiang, Mei et al. "A Real-Time Fall Detection System Based on HMM and RVM." *2013 Visual Communications and Image Processing (VCIP)* (2013): n. pag. Web.

[18] Juang, Chia-Feng, and Chia-Ming Chang. "Human Body Posture Classification By a Neural Fuzzy Network and Home Care System Application." *IEEE Transactions on Systems, Man, and Cybernetics - Part A:*

Systems and Humans IEEE Trans. Syst., Man, Cybern. A 37.6 (2007): 984–994. Web.
[19] Khan, Zafar, and Won Sohn. “Abnormal Human Activity Recognition System Based on R-Transform and Kernel Discriminant Technique for Elderly Home Care.” *IEEE Transactions on Consumer Electronics IEEE Trans. Consumer Electron.* 57.4 (2011): 1843–1850. Web.
[20] Li, Yun, K. C. Ho, and M. Popescu. “A Microphone Array System For Automatic Fall Detection.” *IEEE Transactions on Biomedical Engineering IEEE Trans. Biomed. Eng.* 59.5 (2012): 1291–1301. Web.
[21] Ma, Xin et al. “Depth-Based Human Fall Detection Via Shape Features and Improved Extreme Learning Machine.” *IEEE Journal of Biomedical and Health Informatics IEEE J. Biomed. Health Inform.* 18.6 (2014): 1915–1922. Web.
[22] Ma, Yunqian et al. “Activity Awareness: From Predefined Events to New Pattern Discovery.” *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)* (2006): n. pag. Web.
[23] Mirmahboub, Behzad et al. “Automatic Monocular System For Human Fall Detection Based on Variations in Silhouette Area.” *IEEE Transactions on Biomedical Engineering IEEE Trans. Biomed. Eng.* 60.2 (2013): 427–436. Web.
[24] Nadi, Mai et al. “Fall Detection System of Elderly People Based on Integral Image and Histogram of Oriented Gradient Feature.” *2014 14th International Conference on Hybrid Intelligent Systems* (2014): n. pag. Web.
[25] Nair, Rohit, and Benny Bing. “Intelligent Activity Detection Techniques for Advanced HD Video Surveillance Systems.” *2010 IEEE Sarnoff Symposium* (2010): n. pag. Web.
[26] Ngo, Y T., Hieu V. Nguyen, and Tuan V. Pham. “Study On Fall Detection Based on Intelligent Video Analysis.” *The 2012 International Conference on Advanced Technologies for Communications* (2012): n. pag. Web.
[27] Niżałowska, Katarzyna, Łukasz Burdka, and Urszula Markowska-Kaczmar. “Indoor Head Detection and Tracking on RGBD Images.” *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems* (2014): n. pag. Web.
[28] “A Novel Methodology For Assessing the Fall Risk Using Low-Cost and Off-the-Shelf Devices.” *IEEE Transactions on Human-Machine Systems IEEE Trans. Human-Mach. Syst.* 44.3 (2014): 406–415. Web.
[29] Ozcan, Koray et al. “Automatic Fall Detection And Activity Classification by a Wearable Embedded Smart Camera.” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems IEEE J. Emerg. Sel. Topics Circuits Syst.* 3.2 (2013): 125–136. Web.
[30] Qian, Huimin et al. “Home Environment Fall Detection System Based on a Cascaded Multi-SVM Classifier.” *2008 10th International Conference on Control, Automation, Robotics and Vision* (2008): n. pag. Web.
[31] Rezaee, Khosro, Javad Haddadnia, and Ahmad Delbari. “Intelligent Detection of the Falls in the Elderly Using Fuzzy Inference System and Video-Based Motion Estimation Method.” *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)* (2013): n. pag. Web.
[32] Rougier, Caroline et al. “Fall Detection From Human Shape and Motion History Using Video Surveillance.” *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)* (2007): n. pag. Web.
[33] Rougier, Caroline et al. “Robust Video Surveillance For Fall Detection Based on Human Shape Deformation.” *IEEE Trans. Circuits Syst. Video Technol. IEEE Transactions on Circuits and Systems for Video Technology* 21.5 (2011): 611–622. Web.
[34] Shany, Tal et al. “Sensors-Based Wearable Systems For Monitoring of Human Movement and Falls.” *IEEE Sensors J. IEEE Sensors Journal* 12.3 (2012): 658–670. Web.
[35] Shieh, Wann-Yun, and Ju-Chin Huang. “Speedup The Multi-Camera Video-Surveillance System for Elder Falling Detection.” *2009 International Conference on Embedded Software and Systems* (2009): n. pag. Web.
[36] Thome, N., S. Miguet, and S. Ambellouis. “A Real-Time, Multiview Fall Detection System: A LHMM-Based Approach.” *IEEE Trans. Circuits Syst. Video Technol. IEEE Transactions on Circuits and Systems for Video Technology* 18.11 (2008): 1522–1532. Web.
[37] Tong, Lina et al. “HMM-Based Human Fall Detection And Prediction Method Using Tri-Axial

Accelerometer.” *IEEE Sensors J. IEEE Sensors Journal* 13.5 (2013): 1849–1856. Web.

[38] Vaidehi, V. et al. “Video Based Automatic Fall Detection in Indoor Environment.” *2011 International Conference on Recent Trends in Information Technology (ICRTIT)* (2011): n. pag. Web.

[39] Yu, M. et al. “One Class Boundary Method Classifiers for Application in a Video-Based Fall Detection System.” *IET Comput. Vis. IET Computer Vision* 6.2 (2012): 90. Web.

[40] Yu, Miao et al. “Fall Detection for the Elderly in a Smart Room by Using an Enhanced One Class Support Vector Machine.” *2011 17th International Conference on Digital Signal Processing (DSP)* (2011): n. pag. Web.

[41] Yu, Miao, Syed Mohsen Naqvi, and Jonathon Chambers. “Fall Detection in the Elderly by Head Tracking.” *2009 IEEE/SP 15th Workshop on Statistical Signal Processing* (2009): n. pag. Web.

[42] Yu, Miao et al. “An Online One Class Support Vector Machine-Based Person-Specific Fall Detection System For Monitoring an Elderly Individual in a Room Environment.” *IEEE Journal of Biomedical and Health Informatics IEEE J. Biomed. Health Inform.* 17.6 (2013): 1002–1014. Web.

[43] Yu, Miao et al. “A Posture Recognition-Based Fall Detection System For Monitoring an Elderly Person in a Smart Home Environment.” *IEEE Transactions on Information Technology in Biomedicine IEEE Trans. Inform. Technol. Biomed.* 16.6 (2012): 1274–1286. Web.

[44] Zambanini, Sebastian, Jana Machajdik, and Martin Kampel. “Detecting Falls at Homes Using a Network of Low-Resolution Cameras.” *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine* (2010): n. pag. Web.

[45] Zigel, Y., D. Litvak, and I. Gannot. “A Method For Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound;Proof of Concept on Human Mimicking Doll Falls.” *IEEE Transactions on Biomedical Engineering IEEE Trans. Biomed. Eng.* 56.12 (2009): 2858–2867. Web.

Curriculum Vitae

Robert Gripentog

Howard R. Hughes College of Engineering
Department of Computer Science
University of Nevada, Las Vegas
4505 S Maryland Pkwy, Las Vegas, NV 89154
Office Phone: (702)-895-2507
E-mail address: Robert.Gripentog@unlv.edu

Education

University of Nevada, Las Vegas
Master of Science in Computer Science, Fall 2015
Bachelor of Science – Economics, Fall 2006